

SDK深入专项测试

张迪

关于我



张迪

2017.03~至今 美团点评，平台测试组

2013.07~2017.03 58赶集，用户端测试组

2009.09~2013.07 南开大学，计算机科学与技术

概念简介

功能测试

专项测试

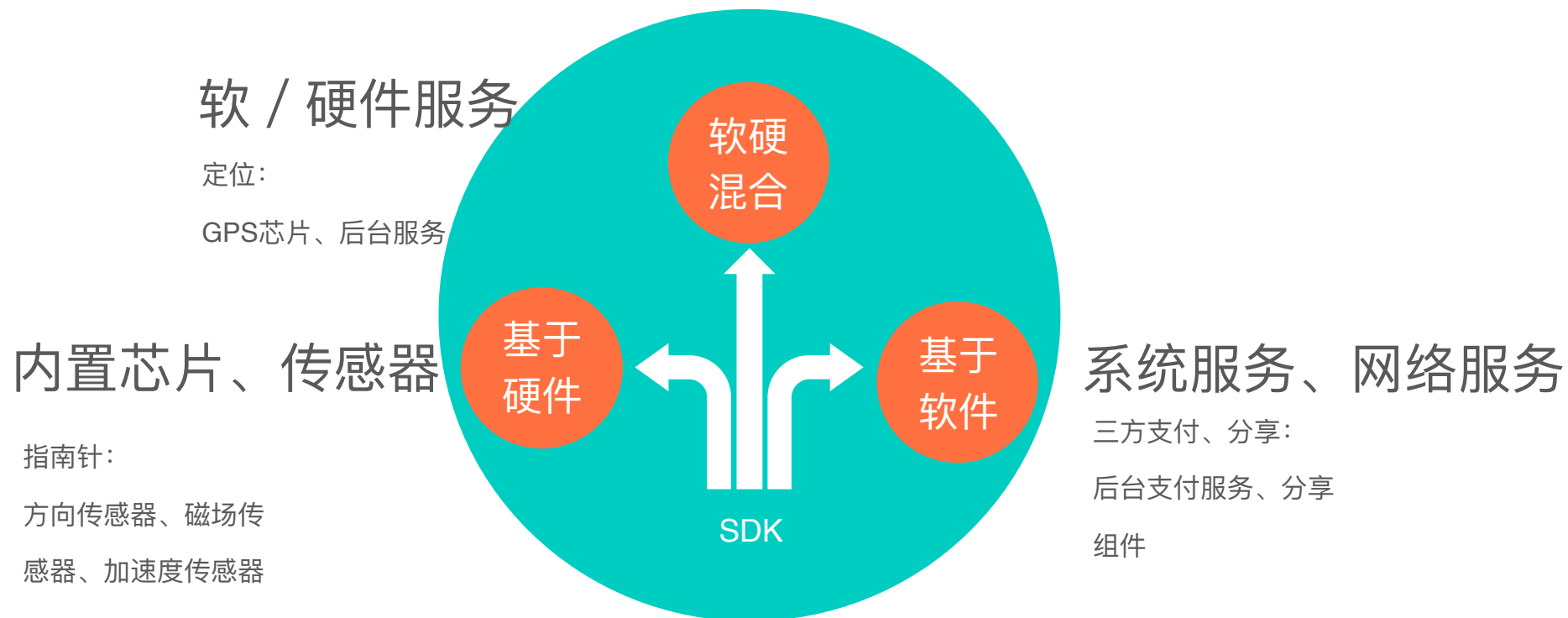
质保流程



什么是SDK

什么是Android SDK

什么是Android第三方SDK





SDK内部构造是什么

Activity

处理UI相关业务

A

B

BraodCast Receiver

一对多通信方式

Service

后台处理耗时任务

S

C

Content Provider

数据共享和操作

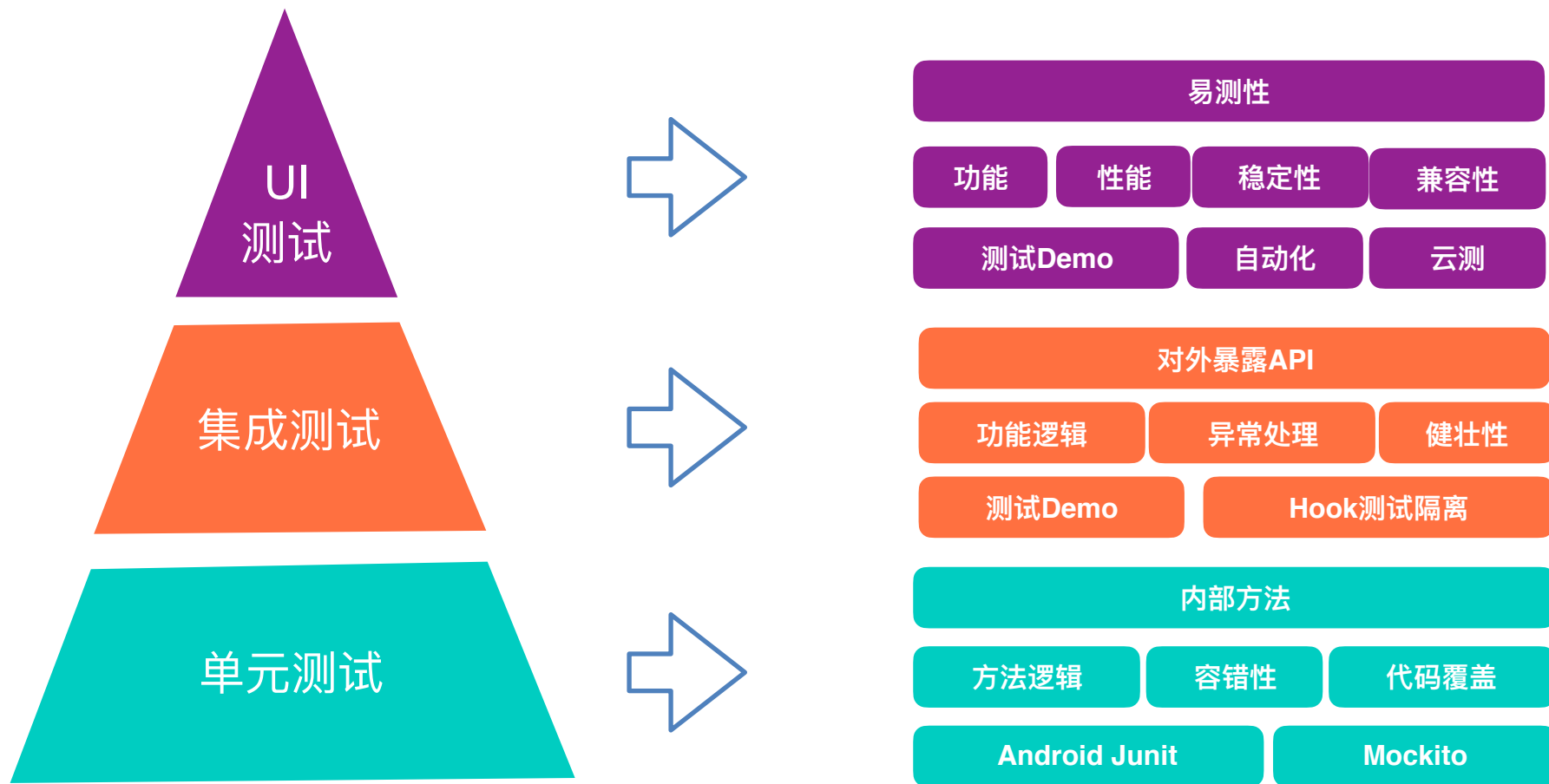
四大组件

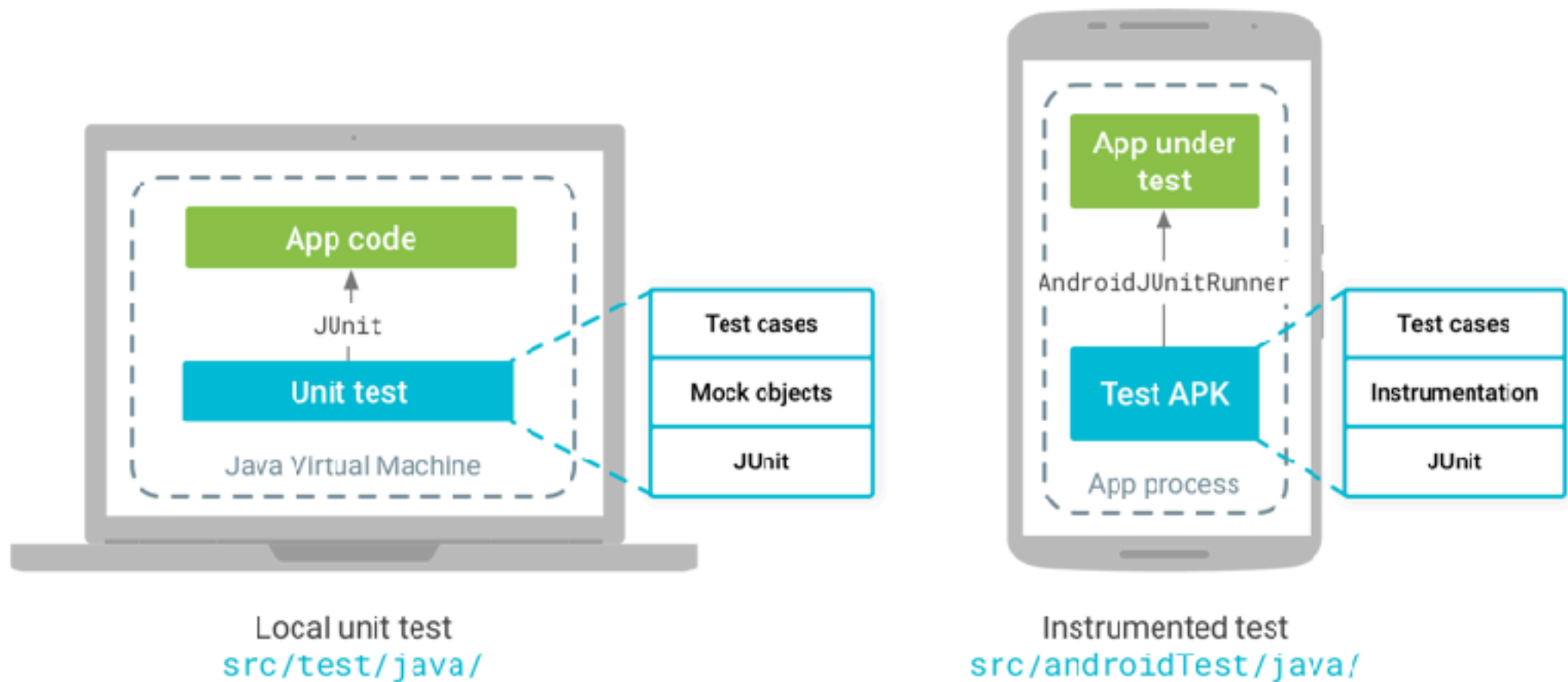
概念简介

功能测试

专项测试

质保流程





测试类型	运行环境	被测目标	TestRunner
本地单元测试	本地JVM	Method	JUnit4
仪器单元测试	真机或模拟器	Method、Activity、Service、Content Provider	AndroidJUnit4

SDK单元测试

前方高能预警：一大波代码来袭！



```
public static boolean getGpsStart(Context context) {  
    if(context != null) {  
        SharedPreferences sp = context.getSharedPreferences("locate", Context.MODE_PRIVATE);  
        return sp.getBoolean("gpsStart", true);  
    }  
    return true;  
}
```

—————→ 从SharedPreferences中读取“gpsStart”对应的值

依赖：

```
dependencies {  
    testCompile 'junit:junit:4.12'  
    testCompile 'org.mockito:mockito-core:1.10.19'  
}
```

JUnit4+Mockito示例

```
@RunWith(MockitoJUnitRunner.class)  // 测试运行器
public class LocationUtilsTest {
    @Mock
    private Context context;  // mock Context

    @Test
    public void getStartTest() {
        // mock SharedPreferences
        SharedPreferences mMockSharedPreferences = mock(SharedPreferences.class);

        // 设置mock 场景
        when(context.getSharedPreferences("locate", Context.MODE_PRIVATE)).thenReturn(mMockSharedPreferences);
        when(mMockSharedPreferences.getBoolean("gpsStart", true)).thenReturn(true);

        // 验证mock场景返回
        Assert.assertEquals(mMockSharedPreferences.getBoolean("gpsStart", true), true);

        // 验证mock对象方法被调用
        verify(mMockSharedPreferences).getBoolean("gpsStart", true);

        // 验证mock对象方法未被调用
        verify(context, never()).getSharedPreferences("locaxxxte", Context.MODE_PRIVATE);

        // 断言被测方法getGpsStart调用及返回
        Assert.assertTrue(LocationUtils.getGpsStart(context)) Hint: double-click to select code
    }
}
```

Espresso示例

```
dependencies {  
    // Other dependencies ...  
    androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2.2'  
}
```

```
@RunWith(AndroidJUnit4.class) → 测试运行器  
@LargeTest  
public class ChangeTextBehaviorTest {  
  
    public static final String STRING_TO_BE_TYPED = "Espresso";  
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(  
        MainActivity.class); → 实例化ActivityTestRule  
  
    @Test  
    public void changeText_sameActivity() {  
        // Type text and then press the button.  
        onView(withId(R.id.editTextUserInput)) → 元素定位  
            .perform(typeText(STRING_TO_BE_TYPED), closeSoftKeyboard());  
        onView(withId(R.id.changeTextBt)).perform(click()); → 操作动作  
  
        // Check that the text was changed.  
        onView(withId(R.id.textToBeChanged)).check(matches(withText(STRING_TO_BE_TYPED)));  
        → 验证  
    }  
}
```

UI Automator示例

```
dependencies {  
    androidTestCompile 'com.android.support.test:uiautomator-v18:2.1.1'  
}
```

```
@RunWith(AndroidJUnit4.class)  
@SdkSuppress(minSdkVersion = 18)  
public class ChangeTextBehaviorTest {  
    private static final String BASIC_SAMPLE_PACKAGE = "com.example.android.testing.uiautomator.BasicSample";  
    private static final int LAUNCH_TIMEOUT = 5000;  
    private static final String STRING_TO_BE_TYPED = "UiAutomator";  
    private UiDevice mDevice;  
    @Before  
    public void startMainActivityFromHomeScreen() {  
        mDevice = UiDevice.getInstance(InstrumentationRegistry.getInstrumentation());  
        mDevice.pressHome();  
        final String launcherPackage = getLauncherPackageName();  
        assertThat(launcherPackage, notNullValue());  
        mDevice.wait(Until.hasObject(By.pkg(launcherPackage).depth(0)), LAUNCH_TIMEOUT);  
        Context context = InstrumentationRegistry.getContext();  
        final Intent intent = context.getPackageManager().getLaunchIntentForPackage(BASIC_SAMPLE_PACKAGE);  
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK); // Clear out any previous instances  
        context.startActivity(intent);  
        mDevice.wait(Until.hasObject(By.pkg(BASIC_SAMPLE_PACKAGE).depth(0)), LAUNCH_TIMEOUT);  
    }  
    @Test  
    public void testChangeText_sameActivity() {  
        mDevice.findObject(By.res(BASIC_SAMPLE_PACKAGE, "editTextUserInput")).setText(STRING_TO_BE_TYPED);  
        mDevice.findObject(By.res(BASIC_SAMPLE_PACKAGE, "changeTextBt")).click();  
        UiObject2 changedText = mDevice.wait(Until.findObject(By.res(BASIC_SAMPLE_PACKAGE, "textToBeChanged")),  
        assertThat(changedText.getText(), is(equalTo(STRING_TO_BE_TYPED)));  
    }  
}
```



```
public class LocalService extends Service {
    // Used as a key for the Intent.
    public static final String SEED_KEY = "SEED_KEY";
    // Binder given to clients
    private final IBinder mBinder = new LocalBinder();
    // Random number generator
    private Random mGenerator = new Random();
    private long mSeed;
    @Override
    public IBinder onBind(Intent intent) {
        // If the Intent comes with a seed for the number generator, apply it.
        if (intent.hasExtra(SEED_KEY)) {
            mSeed = intent.getLongExtra(SEED_KEY, 0);
            mGenerator.setSeed(mSeed);
        }
        return mBinder;
    }
    public class LocalBinder extends Binder {
        public LocalService getService() {
            // Return this instance of LocalService so clients can call public methods.
            return LocalService.this;
        }
    }
    //Returns a random integer in [0, 100].
    public int getRandomInt() {
        return mGenerator.nextInt(100);
    }
}
```

种子数

随机数生成器

绑定Service

被测方法getRandomInt

测试Service示例

```
@RunWith(AndroidJUnit4.class)  // 测试运行器
public class LocalServiceTest {
    @Rule
    public final ServiceTestRule mServiceRule = new ServiceTestRule();  // 实例化Service Rule

    @Test
    public void testWithBoundService() throws TimeoutException {
        // Create the service Intent.  // 实例化Service Intent
        Intent serviceIntent = new Intent(InstrumentationRegistry.getTargetContext(), LocalService.class);

        // Data can be passed to the service via the Intent.  // 将种子数存入Intent
        serviceIntent.putExtra(LocalService.SEED_KEY, 42L);

        // Bind the service and grab a reference to the binder.  // 拿到binder
        IBinder binder = mServiceRule.bindService(serviceIntent);

        // Get the reference to the service, or you can call public methods on the binder directly.  // 利用binder获取被测Service
        LocalService service = ((LocalService.LocalBinder) binder).getService();

        // Verify that the service is working correctly.  // 调用被测方法并断言结果
        assertThat(service.getRandomInt(), is(any(Integer.class)));
    }
}
```

测试覆盖率

-展示测试覆盖度

-指导用例编写

debugAndroidTest

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.e[REDACTED].li.Activity	<div><div></div></div>	42%	<div><div></div></div>	22%	228	289	650	1,051	108	163	28	46
com.e[REDACTED].li.core	<div><div></div></div>	1%	<div><div></div></div>	0%	103	107	544	552	12	16	2	5
com.e[REDACTED].li.collector	<div><div></div></div>	27%	<div><div></div></div>	13%	52	68	208	294	27	41	6	10
com.e[REDACTED].li.utils	<div><div></div></div>	13%	<div><div></div></div>	0%	64	69	241	263	25	30	3	6
com.e[REDACTED].li.entity	<div><div></div></div>	4%	<div><div></div></div>	0%	123	130	201	212	117	124	4	6
com.e[REDACTED].li.test	<div><div></div></div>	0%	<div><div></div></div>	0%	25	25	71	71	13	13	2	2
com.e[REDACTED].li.Service	<div><div></div></div>	80%	<div><div></div></div>	62%	13	31	29	111	10	27	2	8
Total	8,940 of 11,294	26%	510 of 571	11%	608	719	1,944	2,554	312	414	47	83

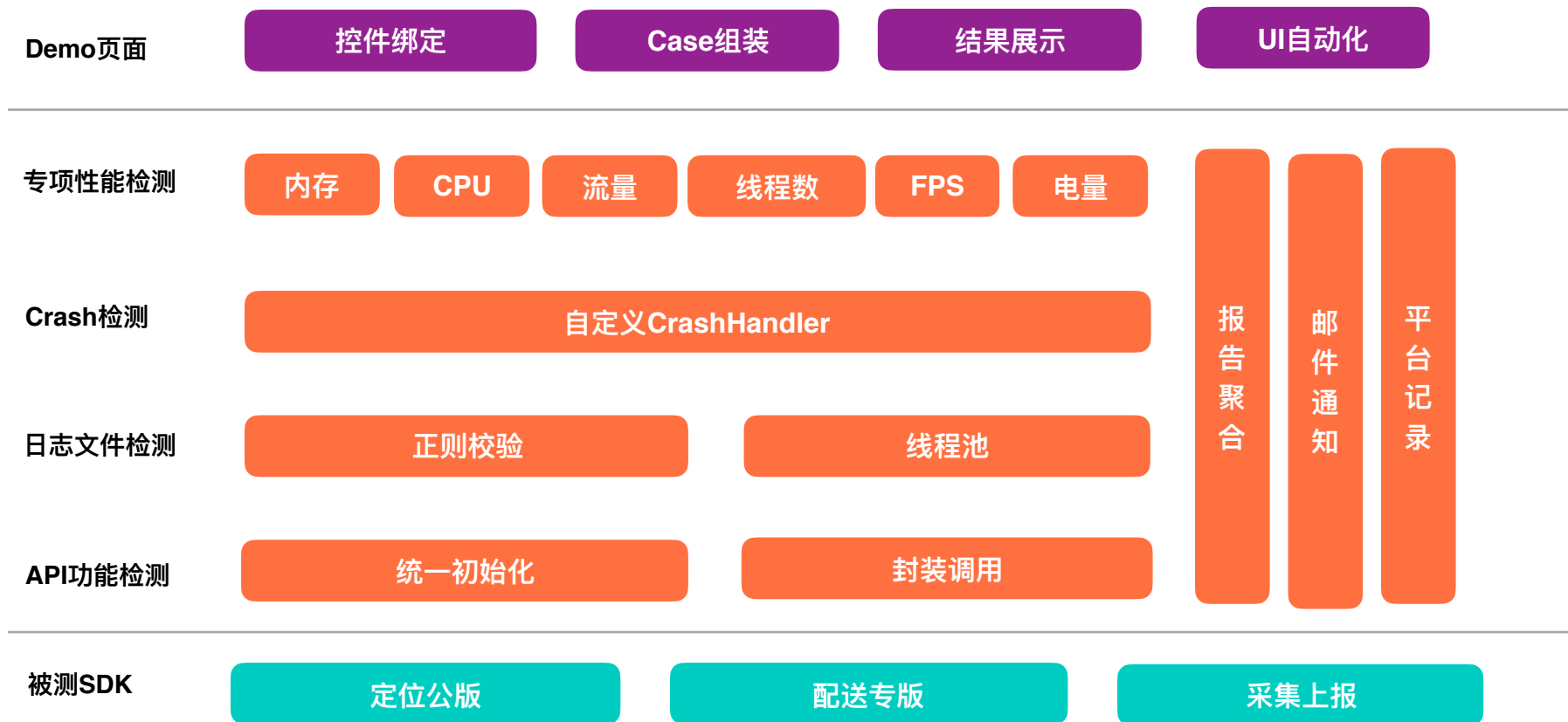
Generated by the Android Gradle plugin 2.3.0

SDK集成测试



SDK集成测试 vs 接口集成测试

	接口集成测试	SDK集成测试
文档准备	接口文档	SDK接入文档
编码能力	RPC Client的实现	基础Android开发
case设计	调用暴露的API 入参：边界值、NULL、异常输入 断言：返回值	调用暴露的API 入参：边界值、NULL、异常输入、系统权限 断言：返回值
测试隔离	mock技术	hook技术



核心步骤:

step 1: 调用被测API

step 2: 设计组装Case

step 3: 返回结果Check

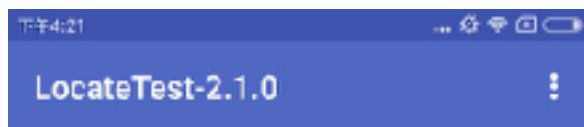




基于SDK日志

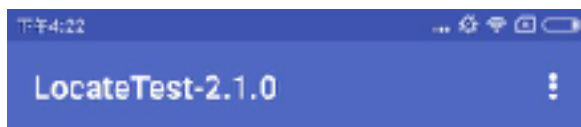
内部关键信息

不易表露现象



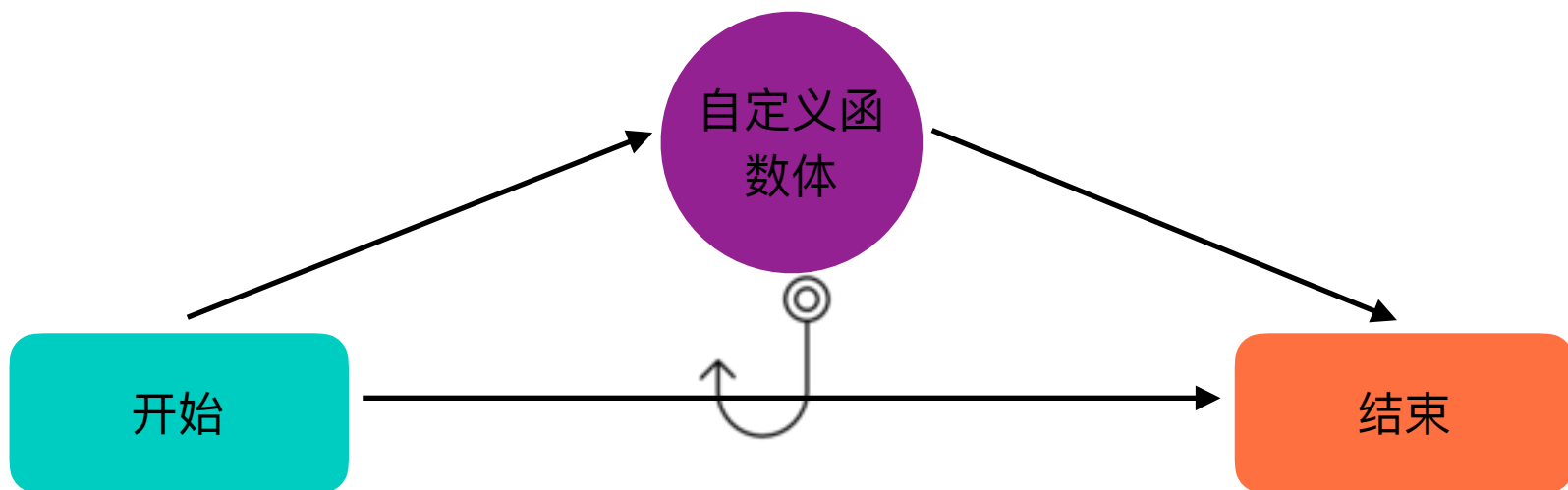
定位SDK测试报告

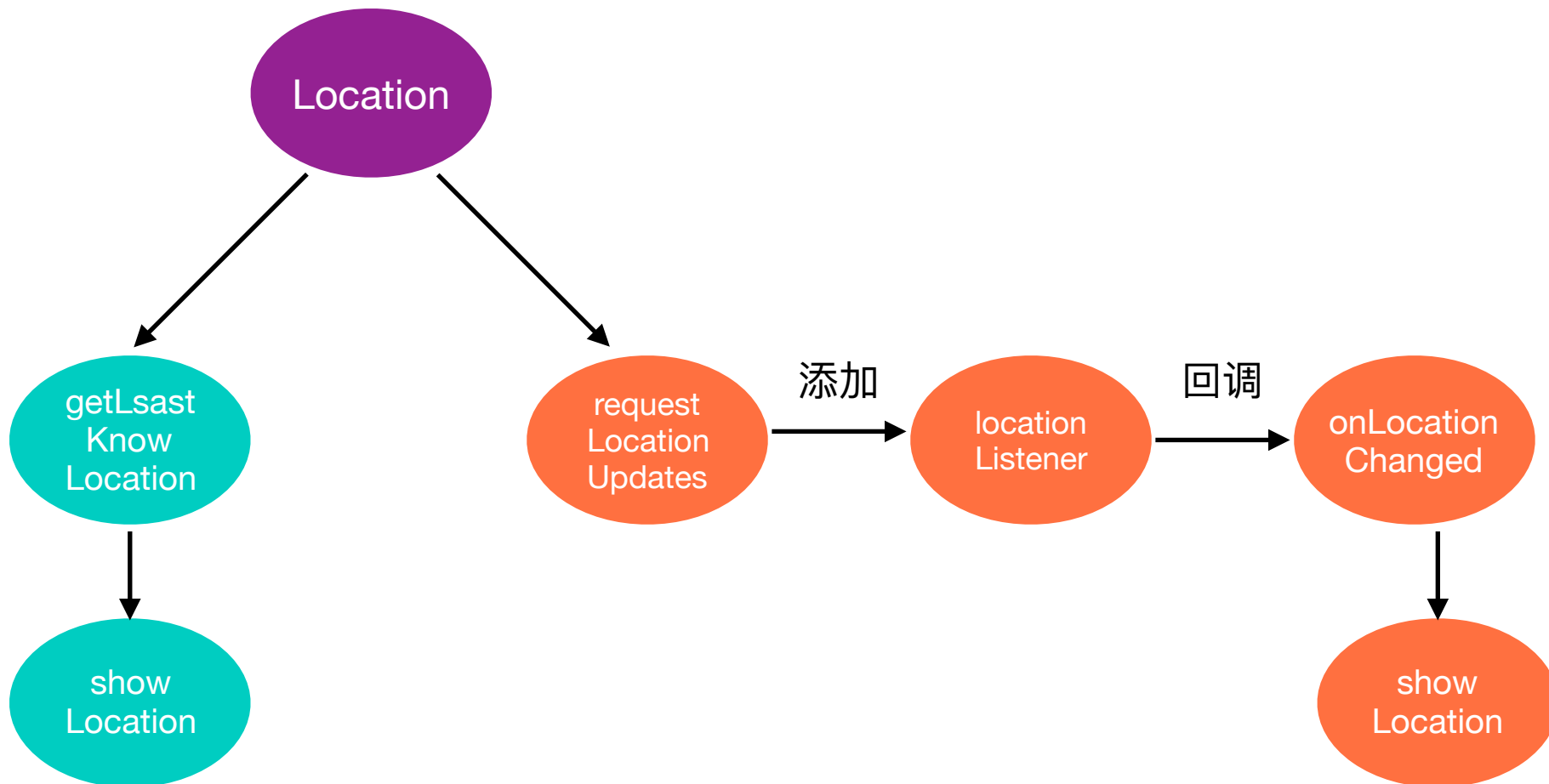
测试结果:	未通过
机型:	MI 5
系统:	7.0
APP:	DEMO
SDK类型:	美团公版
定位版本:	0.2.6_8.8
采集版本:	uid def10
异常检测:	false
GPS定位次数:	14
网络定位次数:	14
线程数:	66
包混淆:	false
RGC:	true



GPS Fix:	17:45:28
测试耗时:	99:59:59
测试距离:	1584.49米
联网类型:	wifi
网络占比:	2G:0/ 3G:0/ 4G:0/ Wi-Fi:100
Wi-Fi扫描:	26次
扫描间隔:	24秒
超时配置:	10秒
联网配置:	已配置
采集更新:	已检测, 无需更新
采集启动:	true
采集上报:	false
采集注入:	true
响应GZip:	true

HOOK(钩子)就是在事件传送到终点前截获并监控事件的传输，像个钩子钩上事件一样，并且能够在钩上事件时，处理一些自己特定的事件





基于Xposed框架Hook GPS信息示例

```
hook_methods("android.location.LocationManager", "getLastKnownLocation", new XC_MethodHook() {
    @Override
    protected void afterHookedMethod(MethodHookParam param) throws Throwable {
        Log.i("jw", "hook getLastKnownLocation...");
        Location l = new Location(LocationManager.PASSIVE_PROVIDER);
        double lo = 66.666d; //经度
        double la = 88.888d; //纬度
        l.setLatitude(la);
        l.setLongitude(lo);
        param.setResult(l);
    }
});
```

概念简介

功能测试

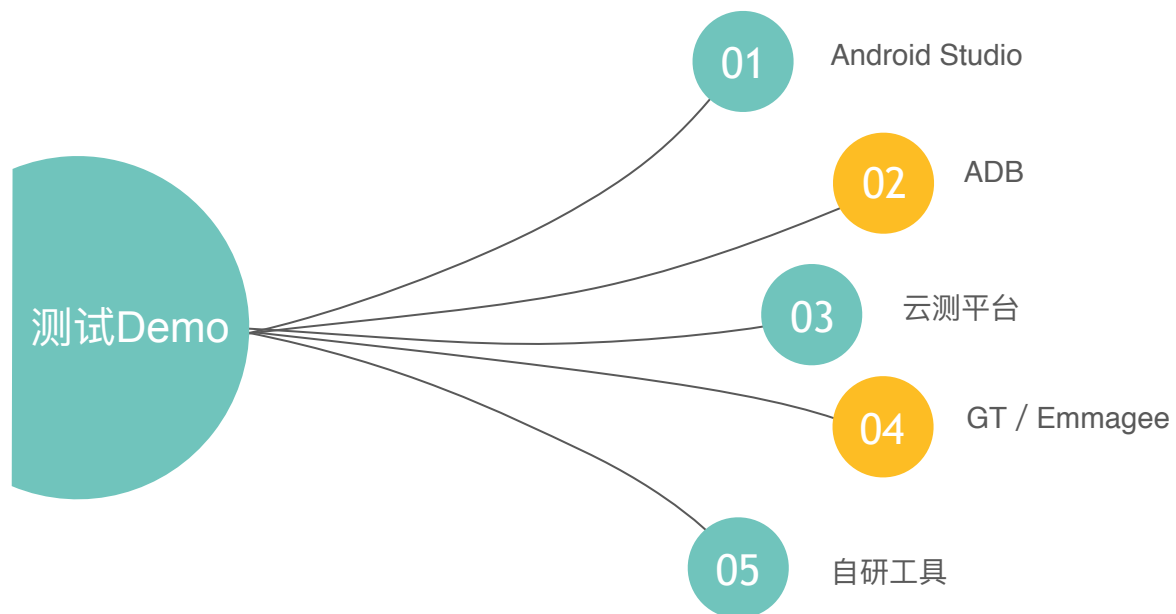
专项测试

质保流程

测什么



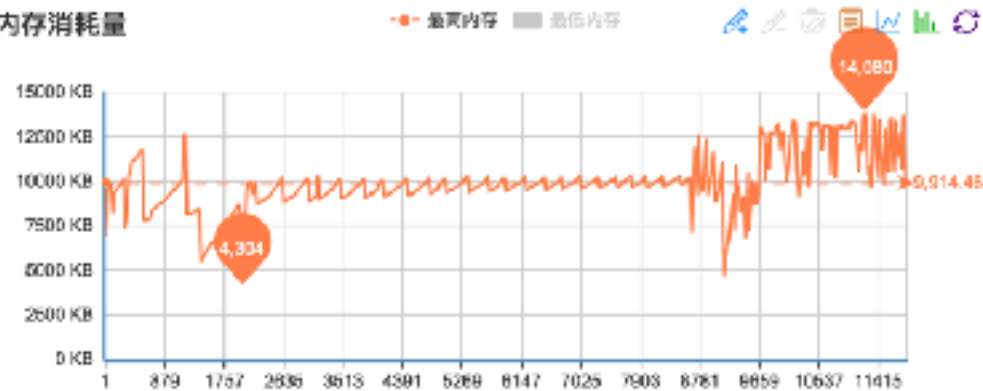
怎么测



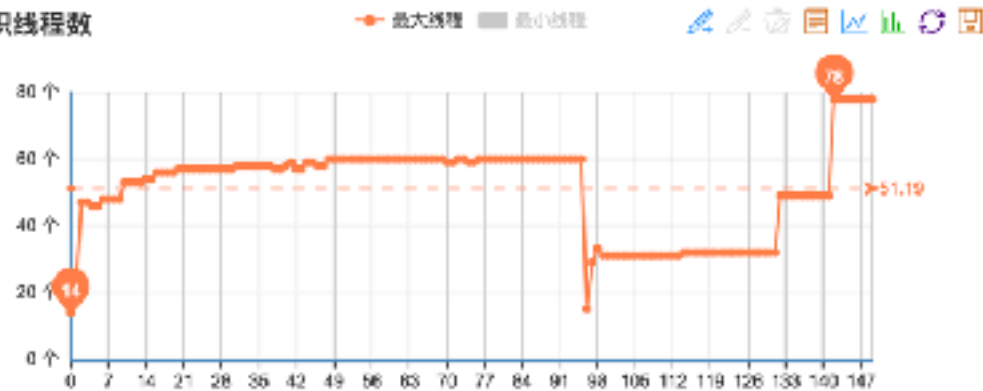
	传统方式	ADB方式	API方式
流量	抓包	<code>adb shell cat /proc/uid_stat/<UID>/tcp_rcv</code> <code>adb shell cat /proc/uid_stat/<UID>/tcp_rcv</code>	<code>android.net.TrafficStats</code> <code>getUidRxBytes(uid)</code> <code>getUidTxBytes(uid)</code>
电量	设置-电量使用统计	<code>adb shell dumpsys batterystats</code>	<code>processAppUsage(SparseArray<UserHandle> asUsers)</code> 注意：未公开API，需要系统权限调用
内存	top	<code>adb shell dumpsys meminfo grep packagename</code>	<code>android.app.ActivityManager</code> <code>getProcessMemoryInfo(pid)</code>
CPU	top	<code>adb shell dumpsys cpuinfo grep packagename</code>	<code>getProcessCPURateByPID(int pid, int interval)</code>
FPS	开发者选项-GPU呈现模式	<code>adb shell dumpsys gfxinfo</code>	<code>getFPSByContext(Context c)</code>

定位SDK性能测试报告

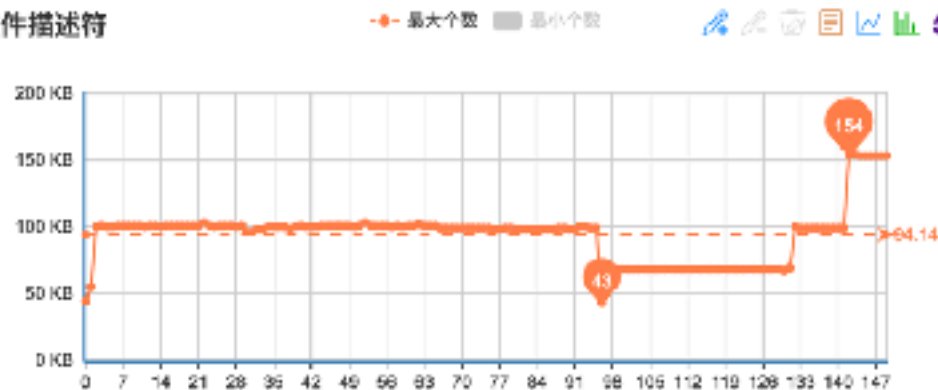
内存消耗量



累积线程数



文件描述符



流量消耗 (KB)



概念简介

功能测试

专项测试

质保流程

SDK提供方

		提测前	测试中	上线后
RD	行动项	准备需求或设计文档 准备SDK接入文档 梳理测试范围重点	过冒烟Case 修改Bug	关注线上Crash平台数据，及时跟进问题
	产出	《需求文档》 《设计文档》 《SDK接入文档》 《测试范围及重点》	《冒烟case通过率》	
QA	行动项	准备测试Case 准备冒烟Case 准备测试Demo 准备测试环境及数据	功能测试 专项性能测试 兼容性测试 稳定性测试	关注线上Crash平台数据，及时跟进问题
	产出	《功能测试Case》 《冒烟Case》 《测试Demo》	《测试报告》	

纬度：

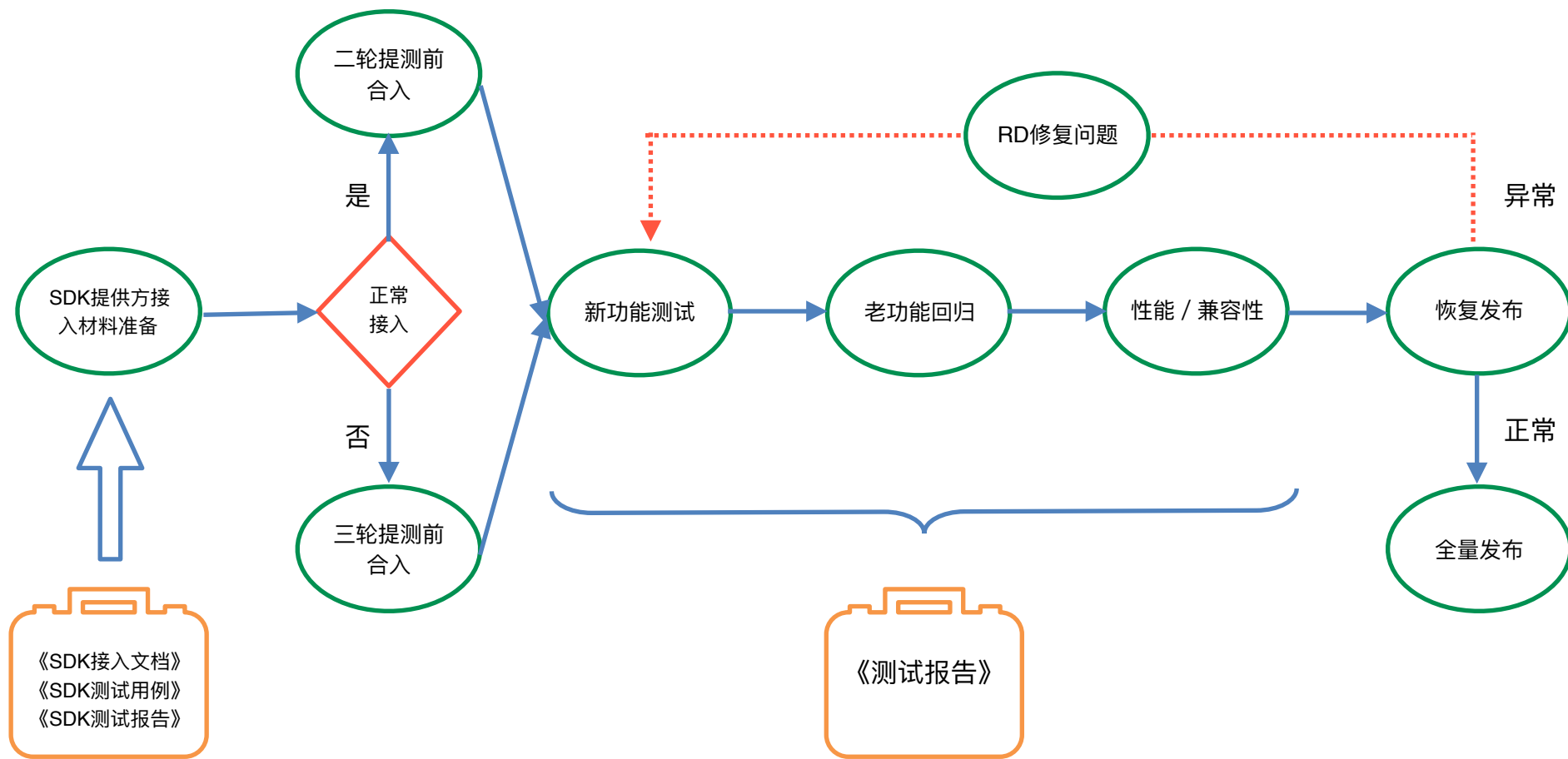
- 通用功能测试报告
- 专项性能测试报告
- 稳定性测试报告
- 兼容性测试报告

细节：

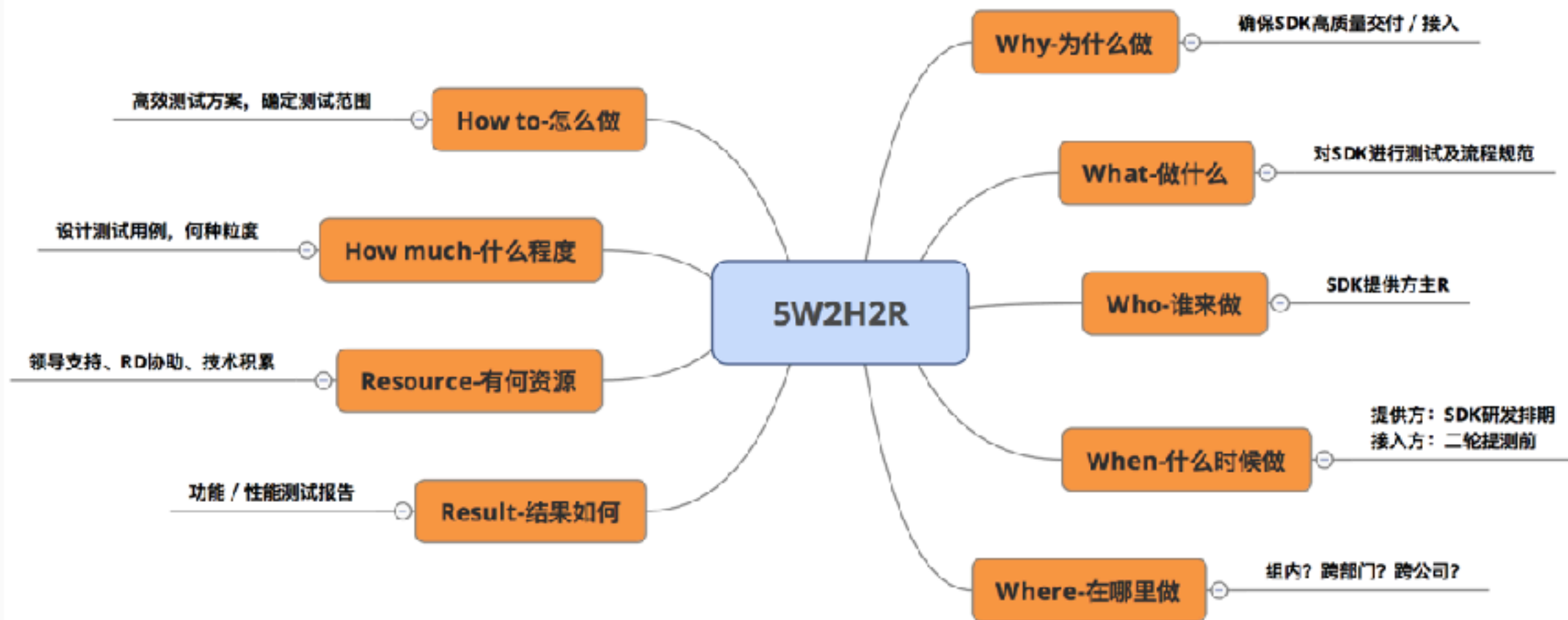
- 测试工具
- 测试方法
- 测试标准
- 测试结果

SDK接入方

注：紧急接入每个Q只能申请一次



5W2H2R



Q&A



扫码关注美团点评技术团队公众号
获取最 IN 的技术资讯

招聘：测试开发工程师
邮箱：
zhangdi15@meituan.com



美团点评 | 技术团队